

## Claims

What is claimed is;

1           1. A method for dynamic compiling, comprising:  
2           loading byte-code on a digital information appliance, said byte-code suitable for  
3           including a tagged section;  
4           identifying the tagged section of the byte-code; and  
5           compiling the tagged section of byte-code;  
6           wherein the tagged section is compiled when the byte-code is loaded so as to enable  
7           the digital information appliance to utilize the tagged section of byte-code without additional  
8           compiling of the tagged section of byte-code by the digital information appliance.

2           2. The method as described in claim 1, further comprising:  
3           encoding application source code to byte-code, the byte-code including code in a  
4           processor-independent form which is suitable for further analysis; and  
5           tagging a section of the byte-code.

1           3. The method as described in claim 2, wherein the byte-code including code in  
2           the processor-independent form is suitable for further analysis including at least one of  
3           suitable for compiler optimization, suitable for processing by interpreters, and suitable for  
4           use in generation of binary instruction for the digital information appliance processing  
5           system.

1           4. The method as described in claim 2, wherein the section of the byte-code is  
2           tagged for being performance sensitive.

1           5. The method as described in claim 1, further comprising storing the compiled  
2           tagged section of byte-code in persistent storage.

1           6. The method as described in claim 1, wherein loading includes validating that  
2 the byte-code conforms with byte-code suitable for utilization by the digital information  
3 appliance.

1 *Am 13*

2

3

4

5

6

7

8

9

10

7. A digital information appliance suitable for dynamic compiling, comprising:  
a processor for implementing a program of instructions; and  
a memory for storing the program of instructions, the program of instructions suitable  
for configuring the digital information appliance to  
load byte-code, said byte-code suitable for including a tagged section;  
identify the tagged section of the byte-code; and  
compile the tagged section of byte-code;  
wherein the tagged section is compiled when the byte-code is loaded so as to enable  
the digital information appliance to utilize the tagged section of byte-code without additional  
compiling of the tagged section of byte-code by the digital information appliance.

1

2

3

4

5

6

7

8

9

10

8. The digital information appliance as described in claim 7, further comprising:  
encoding application source code to byte-code, the byte-code including code in a  
processor-independent form which is suitable for further analysis; and  
tagging a section of the byte-code.

1

2

3

4

5

9. The digital information appliance as described in claim 8, wherein the byte-  
code including code in the processor-independent form is suitable for further analysis  
including at least one of suitable for compiler optimization, suitable for processing by  
interpreters, and suitable for use in generation of binary instruction for a digital information  
appliance's processing system.

1

2

10. The digital information appliance as described in claim 7, wherein the section  
of the byte-code is tagged for being performance sensitive.

1

2

11. The digital information appliance as described in claim 7, further comprising  
storing the compiled tagged section of byte-code in persistent storage.

1           12. The digital information appliance as described in claim 7, wherein loading  
2 includes validating that the byte-code conforms with byte-code suitable for utilization by the  
3 digital information appliance.

09547727.1

13. A system for providing an execution environment that is suitable for dynamic compiling, comprising:

- a memory device suitable for storing computer readable information;
- a loader coupled to the memory device, the loader suitable for loading byte-code to the memory, said byte-code suitable for including a tagged section;
- an identifier coupled to the loader, the identifier suitable for identifying the tagged section of the byte-code;
- a compiler coupled to the identifier;
- wherein the identified tagged section is compiled by the compiler when the byte-code is loaded so as to enable the tagged section of byte-code to be utilized without additional compiling of the tagged section of byte-code.

14. The system as described in claim 13, further comprising:

an encoder for encoding application source code to byte-code, the byte-code including code in a processor-independent form which is suitable for further analysis; and

a tagger for tagging a section of the byte-code.

15. The system as described in claim 14, wherein the byte-code including code in the processor-independent form is suitable for further analysis including at least one of suitable for compiler optimization, suitable for processing by an interpreter, and suitable for use in generation of binary instruction for a processing system.

16. The system as described in claim 14, wherein the section of the byte-code is tagged for being performance sensitive.

17. The system as described in claim 13, wherein the loader includes a validation utility for validating the byte-code conforms with byte-code suitable for utilization by the system.

18. A method for providing an execution environment in an information appliance network, comprising:

- (a) encoding an application source code in a processor independent byte-code;
- (b) tagging at least some portion of said processor independent byte-code; and
- (c) compiling at least some portion of said tagged processor independent byte-code.

19. The execution environment for an information appliance network of claim 18, further including validating at least some portion of said processor independent byte-code.

20. The execution environment for an information appliance network of claim 18, wherein compiling includes identifying the portion of said tagged processor independent byte-code.